Chapter

6

Multivariable Optimization with Constraints

6.1 Formulation of Constrained Optimization

The design variables in an optimization problem cannot be chosen arbitrarily. In some optimization problems, design variables should satisfy some additional specified functional and other requirements. These additional restrictions on design variables are collectively called design constraints.

Most of the constrained optimization problems contain objective function(s) with constraints. The constrained optimization problem can be represented as:

$$\text{Minimize } f(X) \quad X \in \mathbb{R}^n \tag{6.1a}$$

$$\text{subject to } c_i(X) = 0, \quad i \in E \tag{6.1b}$$

$$c_i(X) \geq 0, \quad i \in I \tag{6.1c}$$

where $f(X)$ is the objective function, $c_i(X)$ are constraint functions. E stands for the index set of equations or equality constraints in the problem, I stands for the set of inequality constraints, and both are finite sets. Generally, the constraint equations can be put into different forms for our convenience: such as $c_i(X) \leq b$ turns into $b - c_i(X) \geq 0$. If any point X' (point (2,20) in Fig. 6.1) satisfies all the constraints that point is called a feasible point. The set of all such feasible points is referred to as feasible region (shaded region in Fig. 6.1). Generally, we look for a local (relative) minimum rather than a global minimum. The definition of a constrained local minimizer X^* is that $f(X^*) \leq f(X)$ for all feasible X sufficiently close to X^* [Fletcher, (1986)].

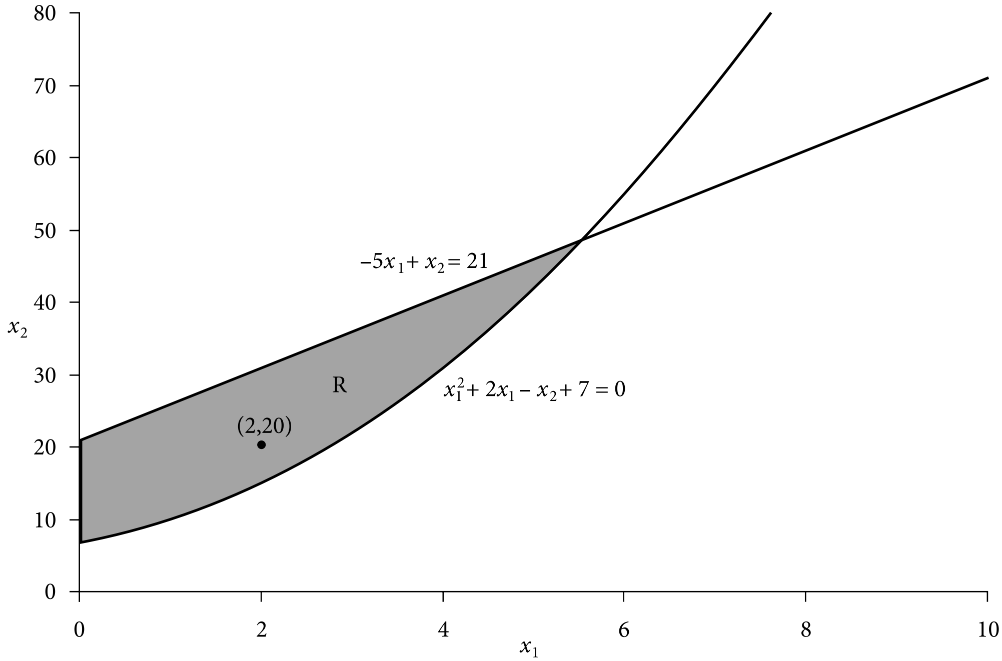

Fig. 6.1 Graphical representation of feasible region

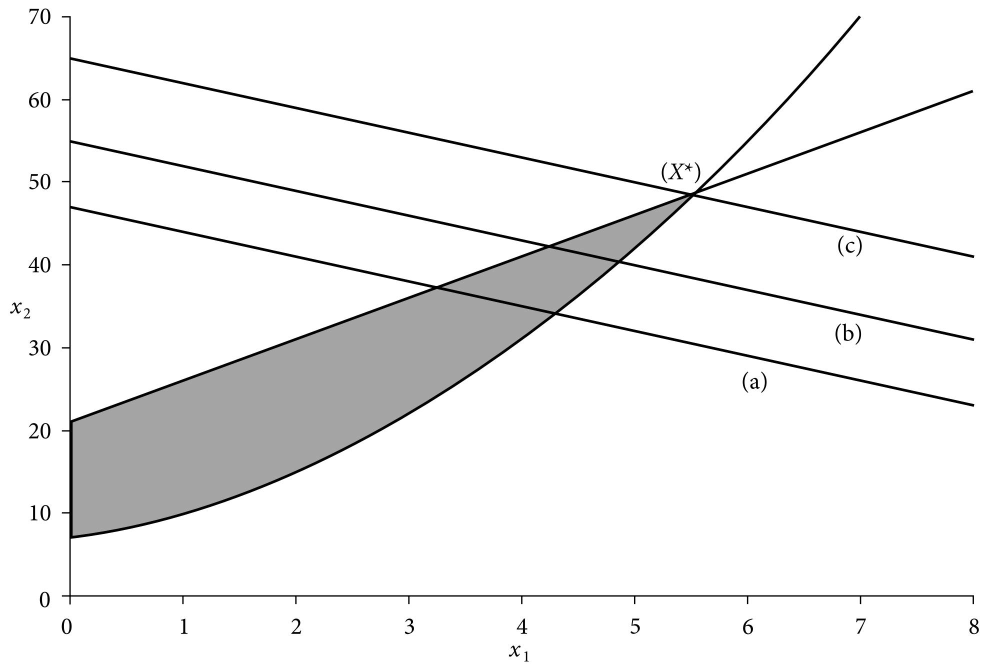

Fig. 6.2 Constrained optimization problem

Figure 6.1 is a graphical representation of a constrained optimization problem. Where R (shaded area) is the feasible region with inequality constraints.

$$x_2 - x_1^2 + 2x_1 \geq 7 \tag{6.2a}$$

$$x_2 - 5x_1 \leq 21 \tag{6.2b}$$

$$x_1 \geq 0 \tag{6.2c}$$

Any objective function that satisfies the constraints (Eq. (6.2a)–(6.2c)) can be optimized by the graphical method. Figure 6.2 explains how maximum value of the objective function $f = 3x_1 + x_2$ (or minimum of $-f$) was obtained. When the line moves from (a) to (b); objective function value (f) increases from 47 to 55. Finally, it moves from (b) to (c) and reaches the maximum point within the feasible region. The maximum value of objective function is 65 at the point (5.5, 48.5).

Depending on the nature of constraints, feasible region may be bounded or unbounded. In Fig. 6.1, the feasible region R is bounded whereas in Fig. 6.3 feasible region is unbounded. If the feasible region is unbounded then the objective function $f(X) \to \mp\infty, X \in R$ or the function may not have a minimum (or maximum) point. The problem also does not provide any solution when the region R is empty; that is when the constraints are inconsistent.

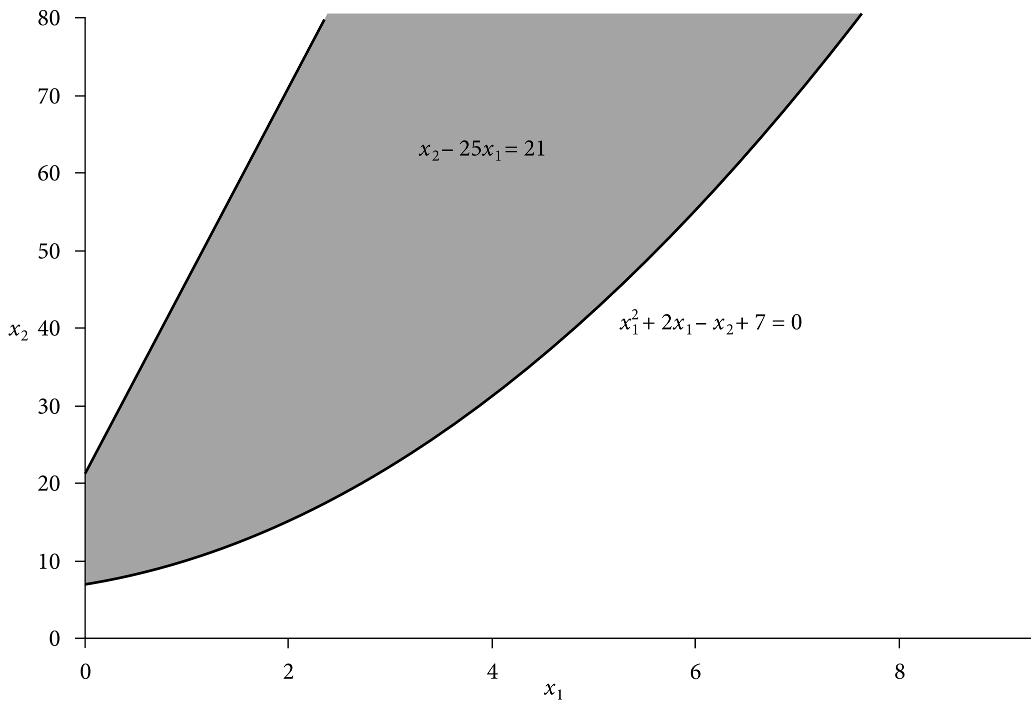

Fig. 6.3 Unbounded feasible region

Constrained multivariable optimization problem can be broadly classified in to two categories; linear programming and nonlinear programming.

6.2 Linear Programming

Linear programming can be viewed as part of great revolutionary development which has given mankind the ability to state general goals and to lay out a path of detailed decisions to take in order to "best" achieve its goals when faced with practical situations of great complexity [Dantzig, 2002]. Linear Programming (LP) is the simplest type of constrained optimization problem. In LPs the objective function $f(X)$ and the constraint functions $c_i(X)$ are all linear functions of decision variables X. The aim of linear programming (LP) is to find out the values of X that minimize a linear objective function subject to a finite number of linear constraints. The linear constraints may be equality and inequality functions of decision variables. The main purpose is to obtain a point that minimizes the objective function and simultaneously satisfies all the constraints. The points that satisfy the constraints are referred as a feasible point. [Chong and Zak (2001)]. This type of optimization problem using linear programming was first developed in 1930s for the optimal allocation of resources. In 1939, L.V. Kantorovich presented several solutions for production and transportation planning problems. T.C. Koopmans contributed significantly during World War II to solve the transportation problems. In 1975, Koopmans and Kantorovich were jointly awarded Nobel Prize in economics for their contribution on the theory of optimal allocation of resources. There are many applications of LP in the field of chemical engineering. Many oil companies are using LP for determining the optimum schedule of product to be produced from the crude oils available. Linear programming also can be used in oil refinery for blending purpose, which optimizes the composition with minimum production cost.

6.2.1 Formulation of linear programming problems

The standard form of a linear programming is as follows:

$$\text{Scalar form: Minimize } f\left(X\right)=c_1x_1+c_2x_2+\ldots+c_nx_n \tag{6.3a}$$

$$\text{subject to} \quad \begin{aligned} a_{11}x_1+a_{12}x_2+\ldots+a_{1n}x_n&=b_1\\ a_{21}x_1+a_{22}x_2+\ldots+a_{2n}x_n&=b_2\\ &\vdots\\ a_{m1}x_1+a_{m2}x_2+\ldots+a_{mn}x_n&=b_m \end{aligned} \tag{6.3b}$$

$$\begin{aligned} x_1&\geq 0\\ x_2&\geq 0\\ &\vdots\\ x_n&\geq 0 \end{aligned} \tag{6.3c}$$

where a_{ij} (i = 1, 2,…,m; j = 1, 2,…,n), b_j, and c_j are known constants, and x_j are called decision variables.

Matrix form: Minimize $f(X) = c^T X$ (6.4a)

subject to $aX = b$ (6.4b)

$X \geq 0$ (6.4c)

where $X = \begin{Bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{Bmatrix}$, $b = \begin{Bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{Bmatrix}$, $c = \begin{Bmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{Bmatrix}$ and $a = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$

Here, a is an $m \times n$ matrix consists of real entries. The value of all b_i should be positive; if any element of b is negative, say the ith element, we have to multiply the ith constraint with –1 to achieve a positive right-hand side. All theorems and solving methods for linear programs are usually written in standard form (Eq. (6.3a)–(6.3c) or Eq. (6.4a)–(6.4c)). All other forms of linear programs can be transformed to the standard form. If a linear programming problem is in the form

Minimize $f(X) = c^T X$ (6.5a)

subject to $aX \geq b$ (6.5b)

$l \leq X \leq u$ (6.5c)

Here, l and u are lower and upper limit of the variable X. The original problem can be converted into the standard form by introducing so-called slack and surplus variables y_i. Algorithms for solving LP problems are divided into two categories (i) simplex method and (ii) non-simplex method

6.2.1.1 Basic LP definitions and results

Now we will simplify the ideas demonstrated in the previous section for problems with 2 to n dimensions. Proofs of the following theorems are not in the scope of this book, it can be found in Dantzig (1963). At the beginning of LPP discussion some standard definitions are given.

Definition 6.1

In an n-dimensional space, a point X has been characterized by an ordered set of n values or coordinates $(x_1, x_2, \ldots, x_n)$. The coordinates of X can also termed as the components of X.

Definition 6.2

If the coordinates of any two points A and B are presented by $x_j^{(1)}$ and $x_j^{(2)}$ ($j = 1, 2, \ldots, n$), the line segment (L) connecting these points is the group of points $X(\lambda)$ whose coordinates are expressed by the equation $x_j = \lambda x_j^{(1)} + (1-\lambda) x_j^{(2)}$, $j = 1, 2, \ldots, n$, with $0 \leq \lambda \leq 1$.

Thus, we can write

$$L = \left\{ X \middle| X = \lambda X^{(1)} + (1-\lambda) X^{(2)} \right\} \tag{6.6}$$

Definition 6.3

A hyperplane in an n-dimensional space is defined as the set of points whose coordinates satisfy a linear equation

$$a_1 x_1 + a_2 x_2 + \ldots + a_n x_n = \mathbf{a}^T X = \mathbf{b} \tag{6.7}$$

A hyperplane H can be described as

$$H(\mathbf{a}, b) = \left\{ X \middle| \mathbf{a}^T X = b \right\} \tag{6.8}$$

Definition 6.4

A feasible solution of the linear programming problem is a vector $X = (x_1, x_2, \ldots, x_n)$, which satisfies Eqs (6.5b) and the bounds (6.5c).

Definition 6.5

A basis matrix is an $m \times n$ nonsingular matrix created from some m columns of the constraint matrix A (since the rank of matrix A rank(A) = m, A possesses at least one basis matrix).

Definition 6.6

A basic solution of a linear program is the unique vector found out by taking a basis matrix, setting each of the n–m variables associated with columns of A not in the basis matrix equal to either l_j or u_j, and solving the resulting square, nonsingular system of equations for the m remaining variables.

Definition 6.7

A basic solution in which all variables satisfy their bounds is called a basic feasible solution (Eq. (6.5c)).

Definition 6.8

A basic feasible solution in which all basic variables x_j are strictly between their bounds, that is, $l_j < x_j < u_j$ is known as a non-degenerate basic feasible solution.

Definition 6.9

An optimal solution is a feasible solution that also minimizes $f(X)$ in Eq. (6.5a).

These definitions give us the following result:

Result 1

The minimum of the objective function $f(X)$ is reachable at a vertex of the feasible region. If minimum value is available at more than one vertex, then the value of the objective function is same at every point of the line segment joining any two optimal vertices.

Result 1 confirms that we need only look at vertices to search for a solution. Therefore, it is our prime concern to know how to characterize vertices algebraically for multi-dimension problems. The next result gives this information.

6.2.1.2 Duality of linear programming

One of the most interesting concepts in linear programming is the duality theory. Every linear programming problem is associated with duality, where there is another linear programming problem with the same data and closely related optimal solutions. These two problems are called to be duals of each other. While one of the problems is called the primal, the other is called dual.

The duality concept is very important due to two main reasons. Firstly, when the primal problem contains a large number of constraints and a smaller number of variables, they can be converted to a dual problem that reduces the computation effort considerably while solving. Secondly, during any decision making in future, the interpretation of the dual variables from the cost or economic point of view seems to be extremely useful.

Primal problem

$$\text{Maximize } f = \sum_{i=1}^{n} c_j x_j \tag{6.9a}$$

$$\text{subject to } \sum_{j=1}^{n} a_{ij} x_j \le b_i, \ (i = 1, 2, \ldots, m) \tag{6.9b}$$

$$x_j \ge 0, \ (j = 1, 2, \ldots, n) \tag{6.9c}$$

The corresponding dual problem can be given by

Dual problem

$$\text{Minimize } z = \sum_{i=1}^{m} b_i y_i \tag{6.10a}$$

$$\text{subject to } \sum_{i=1}^{m} a_{ij} y_i \le c_j, \ (j = 1, 2, \ldots, n) \tag{6.10b}$$

$$y_i \ge 0, \ (i = 1, 2, \ldots, m) \tag{6.10c}$$

Example 6.1

A chemical company produces two chemical products that produced through two different parallel reactions as shown below

$$A + B \xrightarrow{k_1} P_1 \tag{6.11a}$$

$$A + B \xrightarrow{k_2} P_2 \tag{6.11b}$$

The raw materials *A* and *B* have limited supply of 36 kg and 14 kg per day respectively. The reaction 6.11a takes 3 kg *A* and 1kg *B* to produce 1 kg P_1, and the reaction 6.11b takes 2 kg *A* and 1 kg *B* to produce 1 kg P_2. The profit of the company from these products is $14 per kg P_1 and $11 per kg P_2. Formulate a linear programming problem and maximize the daily profit of the company.

Solution

For maximizing the daily profit of the company, we have to formulate the LPP. The objective function (daily profit) can be written as

$$F = 14x_{P_1} + 11x_{P_2} \tag{6.12a}$$

where F is the daily profit, production rates (kg/day) of P_1 and P_2 are x_{P_1} and x_{P_2} respectively.

Constraint functions are formulated based on the limited supply of A and B.

$$\text{Supply of A: } 3x_{P1} + 2x_{P2} \le 36 \tag{6.12b}$$

$$\text{Supply of B: } x_{P1} + x_{P2} \le 14 \tag{6.12c}$$

The production rates (decision variables) can never be negative.

$$x_{P1}, x_{P2} \ge 0 \tag{6.12d}$$

We can write this problem in the similar form of Eq. (6.5a)–(6.5c)

$$\text{Max } F = 14x_{P_1} + 11x_{P_2} \tag{6.12a}$$

subject to

$$3x_{P1} + 2x_{P2} \le 36 \tag{6.12b}$$

$$x_{P1} + x_{P2} \le 14 \tag{6.12c}$$

$$x_{P1}, x_{P2} \geq 0 \tag{6.12d}$$

This LPP can be solved easily by using graphical method. Figure 6.4 gives us an idea about the graphical method. The shaded area is the feasible region, which satisfies the inequality constraints (Eq. (6.12b)–(6.12d)). The objective function F touches the extreme point (8,6) with a value of 178. The solution of the problem is given below:

$F^* = 178$, $x_{P1}^* = 8$, and $x_{P2}^* = 6$

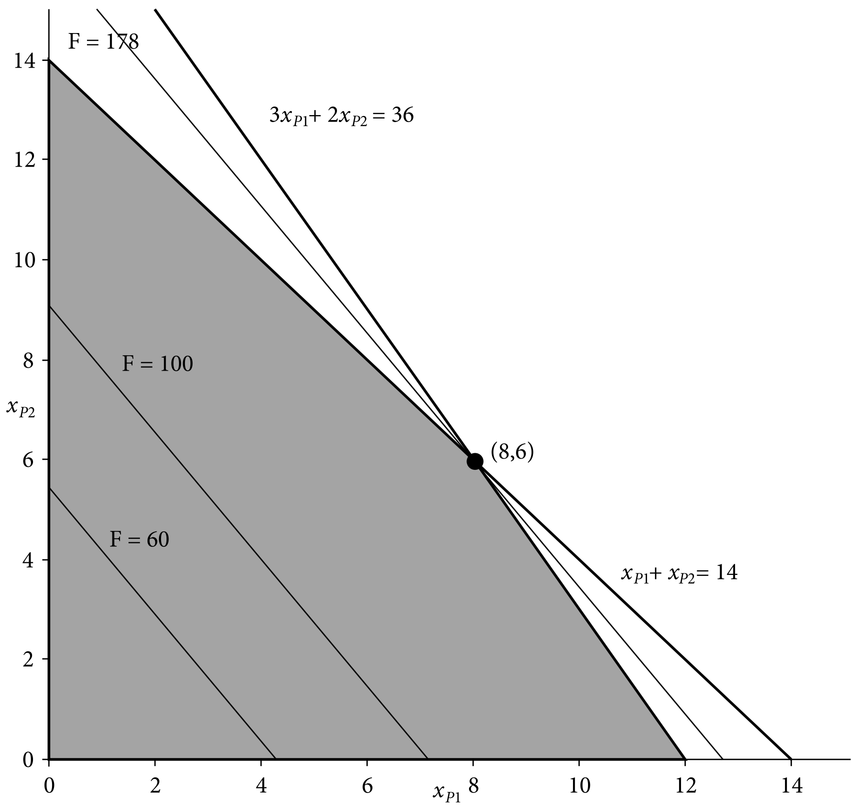

Fig. 6.4 Graphical representation of problem 6.12a–6.12d

6.2.2 Simplex method

For solving linear programming problems, Dantzig [Dantzig, (1963)] developed the simplex method in 1947.It is the perceptive idea behind simplex algorithm that there always exists a vertex of the feasible region, which is an optimal solution to the LP. A vertex is an optimal solution if there is no better neighboring vertex. The terminologies of "vertex", "neighboring vertex" have very clear geometric interpretations. However, these terms are not useful for computer programming. The simplex algorithm is always initiated with an algorithm whose equations are in canonical form. In the first step of the simplex method, certain artificial variables (slack/surplus) are introduced into the standard form. The resulting auxiliary problem is in canonical form. The simplex algorithm consists of a sequence of pivot operations referred to as Phase I that creates a series of different canonical forms. The objective is to find a feasible solution if one exists. If the final canonical form yields such a solution, the simplex algorithm is again applied in a second succession of pivot operations referred to as Phase II.

Algorithm

The simplex method for solving a linear programming problem in standard form produces a series of feasible points $X^{(1)}, X^{(2)}, \ldots$ that terminates to the solution. Each iterate $X^{(k)}$ represents an extreme point and there exists an extreme point at which the solution comes about. Therefore, $n - m$ number of the variables have zero value at $X^{(k)}$ and are referred to as non-basic variables ($N^{(k)}$). The remaining m variables possess a non-negative value, are mentioned to as basic variables ($B^{(K)}$). The simplex method systematically modifies to these sets after each iteration, in order to find the alternative that offers the optimal solution [Fletcher, (1986)]. At each iteration it is convenient to assume that the variables are permuted such that the basic variables are in first m elements of X. Then, we can write $\mathbf{X}^T = \left(\mathbf{X}_B^T, \mathbf{X}_N^T\right)$, where $\mathbf{X}_B^T$ and $\mathbf{X}_N^T$ refer to the basic and non-basic variables respectively. The matrix a in Eq. (6.4b) can also be partitioned similarly into a = $[a_B : a_N]$ where a_B is $m \times m$ basic matrix and a_N is $m \times (n-m)$ non basic matrix. Then, Eq. (6.4b) can be written as

$$\left[\mathbf{a}_B : \mathbf{a}_N\right]\begin{pmatrix}\mathbf{X}_B \\ \mathbf{X}_N\end{pmatrix} = \mathbf{a}_B\mathbf{X}_B + \mathbf{a}_N\mathbf{X}_N = \mathbf{b} \tag{6.13}$$

Since, $\mathbf{X}_N^{(k)} = 0$, it is possible to write

$$\mathbf{X}^{(k)} = \begin{pmatrix}\mathbf{X}_B \\ \mathbf{X}_N\end{pmatrix}^{(k)} = \begin{pmatrix}\hat{\mathbf{b}} \\ 0\end{pmatrix} \tag{6.14}$$

where $\hat{\mathbf{b}} = \mathbf{a}_B^{-1}\mathbf{b}$. Since the basic variables must take non-negative values it is expected that $\hat{\mathbf{b}} \geq 0$.

The partitioning $B^{(k)}$ and $N^{(k)}$ and the extreme point (vertex) $X^{(k)}$ with the above properties ($\mathbf{X}_B^{(k)} = \hat{\mathbf{b}} \geq \mathbf{0}, \mathbf{X}_N^{(k)} = 0$, $\mathbf{a}_B$ non-singular) is referred to as a Basic Feasible Solution (BFS).

From above discussion, an LP in canonical form with m linear constraints and n decision variables, may have a basic feasible solution for every choice of $n-m$ non-basic variables (or equivalently, m basic variables). Therefore, the number of vertices (or BFS) of the feasible region might as well be the same as that of the choice of $n-m$ non-basic variables. How many such choices? From combinatorics, the number of choices are

$$\begin{pmatrix}n \\ n-m\end{pmatrix} = \begin{pmatrix}n \\ m\end{pmatrix} = \frac{n!}{m!(n-m)!} \tag{6.15}$$

Even though, this is a finite number, it could be really large, even if the n, m are relatively small. For example, with n = 20, m = 10, the number of choices are 184756. In practice, the simplex algorithm usually finds the optimal solution after scanning some $4m$ to $6m$ BFS, and very rarely beyond $10m$.

Example 6.2

The steps for a simplex method can be explained through this example

Maximize: $Z = 3x_1 + 2x_2$

subject to constraints

$$2x_1 + x_2 \leq 10$$

$$x_1 + x_2 \leq 8$$

$$x_1 \leq 4$$

and

$$x_1 \geq 0, x_2 \geq 0$$

Solution

We should first write the problem in canonical form by introducing slack variables.

$$2x_1 + x_2 + s_1 = 10$$

$$x_1 + x_2 + s_2 = 8$$

$$x_1 + s_3 = 4$$

in this case, $n = 5$ and $m = 3$ and degree of freedom $n - m = 2$

initialization (finding a starting BFS or a starting vertex): It is easy in this case-just set

NBV = $(x_1, x_2) = (0, 0)$ and BV = $(s_1, s_2, s_3) = (10, 8, 4)$

Optimality test (is the current BFS or vertex optimal?): The current BFS will be optimal if and only if it is better than every neighboring vertex (or every BFS share all but one basic variables). To do this, we try to determine whether there is any way Z can be increased by increasing one of the non-basic variables from its current value zero while all other non-basic variables remain zero (while we adjust the values of the basic variables to continue satisfying the system of equations).

In this case the objective function is Z = $3x_1 + 2x_2$, and Z take value 0 at (0, 0). It is easy to see that no matter we increase x_1 (while holding $x_2 = 0$) or increase x_2 (while holding $x_1 = 0$), we are going to increase Z since all the coefficients are positive. We conclude that the current BFS is not optimal.

Moving to the neighboring BFS (or vertex): Two neighboring BFS share all but one basic variables. In other wards, one of the variables (x_1, x_2) is going to become a basic variable (entering basic variable), and one of (s_1, s_2, s_3) is going to become a non-basic variable (leaving basic variable).

a. Determining the entering basic variable: Choosing an entering basic variable amounts to choosing a non-basic variable to increase from zero. Note $Z = x_1 + 2x_2$. The value of Z is increased by 3 if we increase x_1 by 1, and by 2 if we increase x_2 by 1. Therefore, we choose x_1 as the entering basic variable.

b. Determining how large the entering basic variable can be: We cannot increase the entering variable x_1 arbitrarily, since it may cause some variables to become negative. What is the largest possible value that x_1 can attain? Note x_2 is held at zero. Hence,

$s_1 = 10 - 2x_1 \geq 0$; x_1 cannot exceed 5

$s_2 = 8 - x_1 \geq 0$; x_1 cannot exceed 8

$s_3 = 4 - x_1 \geq 0$; x_1 cannot exceed 4

it follows that the largest x_1 can be is the 4

c. Determine the leaving basic variable: When x_1 takes value 4, s_3 becomes zero. Therefore, s_3 is the leaving basic variable.

Therefore, the neighboring vertex we select is

NBV = (s_3, x_2) BV = (s_1, s_2, x_1)

Pivoting (solving for the new BFS): Recall that we have

$$
\begin{array}{cccccccc}
Z & -3x_1 & -2x_2 & & & & = & 0 & (0) \\
 & 2x_1 & +x_2 & +s_1 & & & = & 10 & (1) \\
 & x_1 & +x_2 & & +s_2 & & = & 8 & (2) \\
 & x_1 & & & & +s_3 & = & 4 & (3)
\end{array}
$$

The goal is to solve for the BFS, and it is going to be achieved by Gaussian elimination. We end up with

$$
\begin{array}{cccccccc}
Z & & -2x_2 & & & +3s_3 & = & 12 & (0) \\
 & & x_2 & +s_1 & & -2s_3 & = & 2 & (1) \\
 & & x_2 & & +s_2 & -s_3 & = & 4 & (2) \\
 & x_1 & & & & +s_3 & = & 4 & (3)
\end{array}
$$

In other wards, each basic variable has been eliminated from all but one row (ith row) and has coefficient +1 in that row. The Gaussian elimination always starts with the row of the leaving basic variable (or the row that achieve the minimal ratio in the preceding step), or the entering basic variable's row is always the row of the leaving basic variable, or the entering basic variable's row is always the row that achieves the minimal ratio in the preceding step.

What we have is that the BFS is

NBV = $(s_3, x_2) = (0, 0)$ BV = $(s_1, s_2, x_1) = (2, 4, 4)$

and

$Z = 12 + 2x_2 - 3s_3$

While, taking value 12 at this BFS.

Iteration The above BFS is not optimal, since we can increase x_2, which increases Z. We do not want to increase S_3, which decreases the value of Z. So the entering basic variable is x_2. How large can x_2 be? Note $s_3 = 0$, we have

$x_2 + s_1 = 2$; x_2 cannot exceed 2

$x_1 + s_2 = 4$; x_2 cannot exceed 4

$x_1 = 4$; no upper bound for x_2

The maximum of x_2 is therefore, 2 achieved at row (1), and the leaving basic variable is the (original) basic variable in row (1), i.e., s_1. In other wards

NBV = (s_1, s_3), BV = (s_2, x_1, x_2) Gaussian elimination yields, starting from row (1) yield,

$$\begin{array}{cccccccc}
Z & & & 2s_1 & & -s_3 & = & 16 & (0) \\
 & & x_2 & +s_1 & & -2s_3 & = & 2 & (1) \\
 & & & -s_1 & +s_2 & +s_3 & = & 2 & (2) \\
 & x_1 & & & & +s_3 & = & 4 & (3)
\end{array}$$

or the new BFS is

NBV = $(s_1, s_3) = (0,0)$, BV = $(s_2, x_1, x_2) = (2,4,2)$.

The value of Z is

$$Z = 16 - 2s_1 + s_3$$

and it attains value 16 at this BFS.

The BFS is still not optimal, and clearly s_3 will be the entering basic variable. Note $s_1 = 0$, we have

$x_2 - 2s_3 = 2$; no upper bound for s_3

$s_2 + s_3 = 2$; s_3 cannot exceed 2

$x_1 + s_3 = 4$; s_3 cannot exceed 4

The maximum of s_3 is therefore, 2 achieved at row (2), and the leaving basic variable is the (original) basic variable in row (1), i.e., s_2. In other wards

NBV = (s_1, s_2), BV = (x_1, x_2, s_3).

the Gaussian elimination yields, starting from row (1) yields,

$$\begin{array}{lllllllll} Z & & & +s_1 & +s_2 & & = & 18 & (0) \\ & & x_2 & -s_1 & +2s_2 & & = & 6 & (1) \\ & & & -s_1 & +s_2 & +s_3 & = & 2 & (2) \\ & x_1 & & +s_1 & -s_2 & & = & 2 & (3) \end{array}$$

or the new BFS is

$$\text{NBV} = (s_1, s_2) = (0,0), \quad \text{BV} = (x_1, x_2, s_3) = (2,6,2).$$

The value of Z is

$$Z = 18 - s_1 - s_3$$

and it attains value 18 at this BFS.

The BFS turns out to be optimal; any increase in the non-basic variable will decrease the value of Z. Hence,

Max $Z = 18$, achieved at $(x_1^*, x_2^*) = (2,6)$.

Degeneracy in Linear Programming

A Linear Programming is degenerate if in a basic feasible solution, one of the basic variables takes on a zero value. Degeneracy is caused by redundant constraint(s) and could cost simplex method extra iterations.

Constraints are either redundant or necessary; redundant constraints are constraints that may be deleted from the set without changing the region defined by the set. It is observed that redundancies exist in most practical problems. The importance of detecting and removing redundancy in a set of linear constraints is the avoidance of all calculations associated with those constraints when solving an associated mathematical programming problem.

Example 6.3

$$\text{Max } Z = 2x_1 + x_2 + x_3 \tag{6.16a}$$

$$\text{subject to: } x_1 + 2x_2 \le 1 \tag{6.16b}$$

$$-x_2 + x_3 \le 0 \tag{6.16c}$$

$$x_1, x_2, x_3 \ge 0 \tag{6.16d}$$

this example shows that $x_2 \ge 0$, follows from constraints $-x_2 + x_3 \le 0$ and $x_3 \ge 0$. Therefore, $x_3 \ge 0$ is redundant constraint.

6.2.3 Nonsimplex methods

Simplex method is extensively used in practice to solve LP problems. However, the computational time required to get a solution using the simplex method increases rapidly when the number of components n of the variable $x \in \mathbb{R}^n$ increases. The major drawback of simplex algorithm is exponential complexity. As the number of component n of the variable x increases, the required time also increases exponentially. This correlation is also called the complexity of the algorithm. Therefore, we can say that the simplex algorithms possesses an exponential complexity. The complexity of the simplex algorithm is often written as $O(2^n - 1)$. For a number of years, exponential complexity and polynomial complexity have been distinguished by the computer scientists. If any algorithm used to solve linear programming problems has polynomial complexity, in that case the time required to find the solution is bounded by a polynomial in n. It is obvious that the polynomial complexity is more desirable than exponential complexity. This concept led to a concern in developing algorithms to solve linear programming problems, which have polynomial complexity. These algorithms are able to find the solution in an amount of time that is restricted by a polynomial in the number of variables. Therefore, the existence of an algorithm to solve linear programming problems with polynomial complexity is an important issue. This issue was somewhat resolved by Khachiyan [L. G. Khachiyan, (1979)], this method is also called the ellipsoid algorithm. Then, Karmarkar's interior point method [N. K. Karmarkar. (1984)] was developed in 1984. In this chapter we will consider two non-simplex methods namely Khachiyan's ellipsoid method and Karmarkar's interior point method.

6.2.3.1 Khachiyan's ellipsoid method

The first polynomial time algorithm for linear programming was developed in 1979 by the Russian mathematician Khachiyan [L. G. Khachiyan, (1979)]. He has shown that linear programs (LPs) can be solved efficiently; more accurately for LP problems that are polynomially solvable. Khachiyan's approach was developed based on ideas which are analogous to the Ellipsoid Method arising from convex optimization [Rebennack (2008)]. The ellipsoid method is an algorithm that finds an optimal solution of LPP in a finite number of steps. This method generates a sequence of ellipsoids; volume of those ellipsoids decreases uniformly at each step, thus, enclosing the minimum of a convex function (see Fig. 6.5).

The Basic Ellipsoid Algorithm

The linear programming problem

$$\text{Optimize } z = c^T x \tag{6.17a}$$

$$\text{subject to } Ax \le b \tag{6.17b}$$

$$x \ge 0 \tag{6.17c}$$

A is an $m \times n$ matrix i.e., $A \in R^{m\times n}$, c is an n-vector, i.e., $c \in R^n$, x is an n-tuple, $x \in R^n$and $b \in R^m$ suppose we are interested to find an n-vector x, satisfying

$$A^T x \le b \tag{6.18}$$

The column of A corresponding to outward drawn normals to the constraints are denoted by $\alpha_1, \alpha_2, \ldots, \alpha_n$ and the components of b denoted by $\beta_1, \beta_2, \ldots, \beta_n$. Thus, Eq. (6.18) can be restated as

$$\alpha_i^T x \le \beta_i,\ i = 1, 2, \ldots, n \tag{6.19}$$

throughout, the calculation we assume that n is greater than one.

The basic ellipsoid iteration

The ellipsoid method construct a sequence of ellipsoids; E_0, E_1, E_2,… E_k,… each of which contains a point satisfying Eq. (6.18), if one exist. E_k is defined as follows:

$$E_k = \left\{ x \in R^n \left| \left(x - x_k\right)^T B_k^{-1} \left(x - x_k\right) \le 1 \right. \right\} \tag{6.20}$$

Where

$$x_{k+1} = x_k - \tau \frac{B_k \alpha}{\sqrt{\alpha^T B_k \alpha}} \tag{6.21}$$

$$B_{k+1} = \delta \left(B_k - \frac{\sigma \left(B_k \alpha\right)\left(B_k \alpha\right)^T}{\alpha^T B_k \alpha} \right) \tag{6.22}$$

$$\tau = \frac{1}{n+1}, \sigma = \frac{2}{n+1}, \text{ and } \delta = \frac{n}{\sqrt{n^2 - 1}} \tag{6.23}$$

τ is known as the step parameter, while δ and σ are the expansion and dilation parameters respectively. On the $(k + 1)$th iteration, the algorithm verifies whether the centre x_k of the current ellipsoid E_k satisfies the constraints Eq. (6.18). If so, the iteration stops. If not some constraints violated by x_k, say

$$\alpha_i^T x \le B_i \tag{6.24}$$

are chosen and the ellipsoid of minimum volume that contains the half ellipsoid:

$$\left\{ x \in E_k \left| \alpha^T x \le \alpha^T x_k \right. \right\} \tag{6.25}$$

constructed.

The new ellipsoid and its centre are represented by E_{k+1} and x_{k+1} respectively as defined above. The iteration is again repeated. Note that the initial ellipsoid, E_0 is arbitrarily assumed.

Apart from the initial ellipsoid, these steps give a possibly infinite iterative algorithm for determining the feasibility of (6.18).

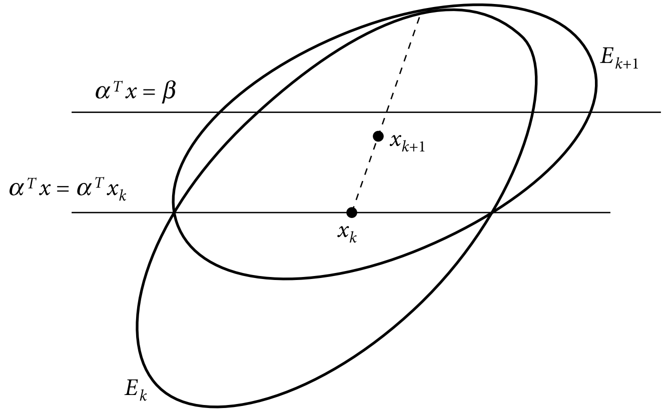

Fig. 6.5 Ellipsoid method

In essence, Khachiyan showed that it is possible to determine whether Eq. 6.18 is feasible or not within a pre-specified (polynomial) number of iterations (i.e., $6n^2L$ iteration) by

a) revising this algorithm to account for finite precision arithmetic

b) applying it to a appropriate perturbation of system

c) selecting E_0 appropriately

System (6.18) is feasible if and only if it terminates with feasible solution of the perturbed system within the stipulated number of iteration [Bland, (1981)].

6.2.3.2 Karmarkar's interior point method

In 1984, Karmarkar [N. K. Karmarkar, (1984)] proposed a new linear programming algorithm that has polynomial complexity. This algorithm is able solve some complex real-world problems such as scheduling, routing and planning more efficiently than the simplex method. This important work of Karmarkar led to the development of many other non-simplex techniques usually referred to as interior point methods.

Unlike the simplex method, the Karmarkar's interior point methods traverse the interior (internal space) of the feasible region as shown in Fig. 6.6. In exterior point method, the search algorithm follows a path A, B, C and reaches the optimum point. Whereas the interior point methods follow the path like 1, 2, 3, 4; then reaches the optimum. The main difficulty with interior point methods is that it needs to identify the best among all feasible directions at a specified solution. The main objective is to reduce the number of iterations to improve computational effort [Ravindran *et al.* (2006)]. It is found that, for a large problem, Karmarkar's method is 50 times faster than the simplex method.

Algorithm

Karmarkar's method is developed based on two observations:

i. When the current solution is close to the centre of the feasible region, we can move along the steepest descent direction that decreases the value of the objective function f by the maximum amount.

ii. Transformation of the solution space is always possible without changing the characteristic of the problem in order to keep the current solution near the centre of the feasible region.

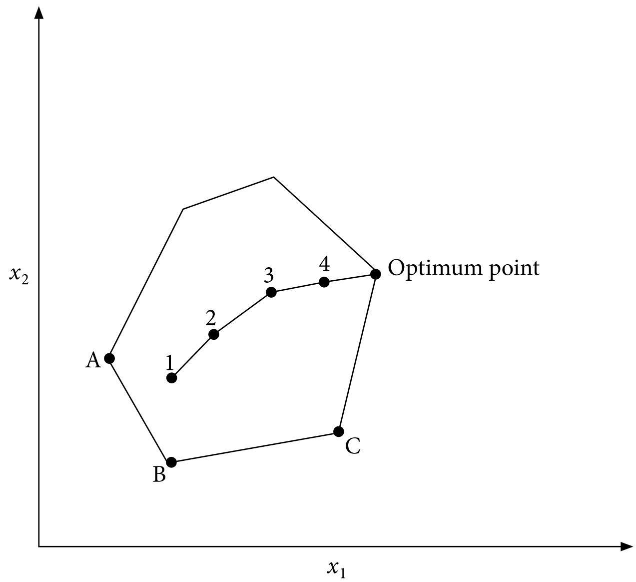

Fig. 6.6 Interior point method

In many numerical problems, the numerical instability can be reduced by changing the units of data or rescaling. Karmarkar noticed that the variables could be transformed in such a way that straight lines remain straight lines whereas distances and angles change for the feasible space. Karmarkar's projective scaling algorithm begins with an interior solution, by transforming the feasible region in a way so that the current solution is positioned at the center of the transformed feasible region. Figure 6.7(a) is the feasible region with the point $X^{(k)}$, far from the center. This region was converted to region 6.7(b), where the point $X'^{(k)}$ is near the center of the region. Where

$$X^{(k)} = \begin{bmatrix} x_1^{(k)} \\ x_2^{(k)} \\ \vdots \\ x_n^{(k)} \end{bmatrix}, \ X'^{(k)} = \begin{bmatrix} \frac{1}{n} \\ \frac{1}{n} \\ \vdots \\ \frac{1}{n} \end{bmatrix} = \frac{e}{n} \quad \text{and } e = \begin{bmatrix} 1 & 1 & \cdots & 1 \end{bmatrix}^T \tag{6.26}$$

After transformation, the searching route follows the steepest descent direction with a step that stops very close to the boundary of the transformed feasible region. Afterward, the improved solution was mapped to the original feasible region by using an inverse transformation. This procedure is repeated until an optimum is achieved with the desired accuracy.

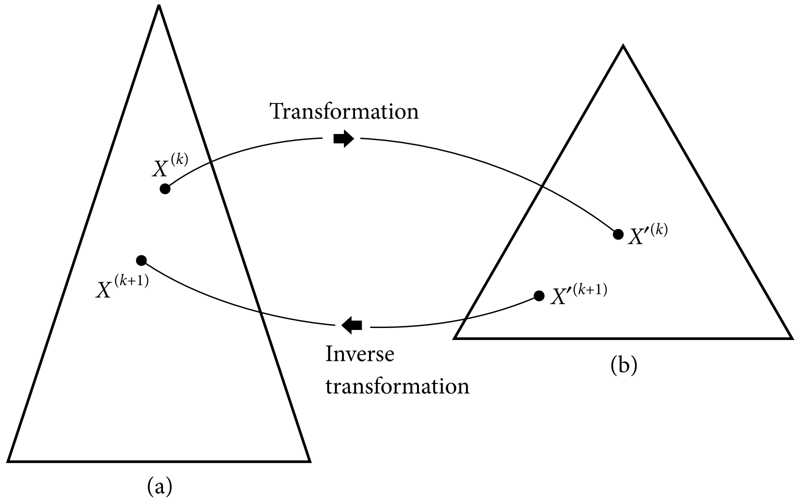

Fig. 6.7 Karmarkar's region inversion

Karmarkar's method employs the LPP in the following form

$$\text{Minimize } f = c^T X \tag{6.27a}$$

subject to

$$[a]X = 0 \tag{6.27b}$$

$$e^T X = 1 \tag{6.27c}$$

$$X \geq 0 \tag{6.27d}$$

where $X = \{x_1\ x_1\ \ldots\ x_n\}^{\mathrm{T}}$, $c = \{c_1\ c_2\ \ldots\ c_n\}^T$, $e = \{1\ 1\ \ldots 1\}^T$, and $[a]$ is an $m \times n$ matrix. Beside this, an interior feasible starting solution to Eq. (6.27b)–(6.27d) must be known. Usually, $X = \left\{\frac{1}{n}\ \ \frac{1}{n}\ \ \cdots\ \ \frac{1}{n}\right\}^T$ is preferred as the starting point. The optimum value of the objective function f must be zero for the problem. Therefore,

$$X^{(1)} = \left\{\frac{1}{n}\ \ \frac{1}{n}\ \ \cdots\ \ \frac{1}{n}\right\}^T = \text{interior feasible} \tag{6.28}$$

$$f_{\min} = 0 \tag{6.29}$$

Although, most LPPs may not be available in the form of Eq. (6.27b)–(6.27d) while satisfying the conditions of Eq. (6.28), it is possible to put any LPP in a form that satisfies Eq. (6.27b)–(6.27d) and (6.28) as indicated below. The process of this transformation is discussed below. Karmarkar used "Projective Transformations" which are non-linear transformations under which lines and subspaces are preserved but distances and angles are distorted [Lemire, (1989)].

The algorithm creates a sequence of points $X^{(1)}, X^{(2)}, \ldots, X^{(k)}$. The whole process can be divided into 4 steps:

Step 1 Initialization $X^{(1)}$ = center of the simplex

Step 2 Computation of the next point in the sequence

$$X^{(k+1)} = \varphi\left(X^{(k)}\right) \tag{6.30}$$

Step 3 Checking for infeasibility

Step 4 Checking for optimality
go to Step 1.

Now we will discuss the steps in detail

1. The function $b = j(a)$ can be defined by the following sequence of operations.
   Let $D = \text{diag}\{a_1\ a_2 \ldots a_n\}$ be a diagonal matrix whose ith diagonal entry is a_i.
   Let

$$B = \left[\frac{AD}{e^T}\right] \tag{6.31}$$

   i.e., augment the matrix AD with a row of all l's. This guarantees that $KerB$
2. Compute the orthogonal projection of Dc into the null space of B.

$$c_p = \left[I - B^T\left(BB^T\right)^{-1} B\right]Dc \tag{6.32}$$

3. $\hat{c} = \dfrac{c_p}{|c_p|}$ i.e., $\hat{c}$ is the unit vector in the direction of c_p.

4. $b' = a_0 - \alpha r\hat{c}$ i.e., take a step of length αr in the direction $\hat{c}$, where r is the radius of largest inscribed sphere. By the Euclidean distance formula:

$$r = \frac{1}{\sqrt{n(n-1)}} \tag{6.33}$$

   and $\alpha \in (0,1)$ is a parameter which can be set equal to 1/4. The value of α $(0 < \alpha < 1)$ ensures that all iterates are interior points.
5. Apply inverse projective transformation to b'

$$b = \frac{Db'}{e^T Db'} \tag{6.34}$$

Return b.

Step 3. Check for in feasibility.

A "potential" function can be defined by

$$f(X) = \sum_i \ln \frac{c^T X}{x_i} \tag{6.35}$$

At each step, a certain improvement δ in the potential function is expected. The value of δ depends on how the parameter α is selected in the above Step. For example, if value of $\alpha = 1/4$ then $\delta = 1/8$. If the expected improvement is not found, i.e., if $f(X^{(k+1)}) > f(X^{(k)}) - \delta$ then we stop and conclude that the minimum value of the objective function must be strictly positive. The standard linear program problem can be transformed to the canonical form, then this condition corresponds to the situation that the original problem does not possess a finite optimum i.e., it is either unbounded or infeasible.

Step 4. Check for optimality

The optimality check should be done periodically. It includes moving from the current interior point to an extreme point without increasing the objective function value and checking the extreme point for optimality. This process is followed only when the time spent since the last check exceeds the time required for checking [Karmarkar, (1984)].

6.2.4 Integer linear programming

The linear programming problems that have been discussed so far are all continuous, in the sense that all decision variables are permitted to be fractional. For instance, we might optimize rate of production, flow rate of reactant and time of operation for batch process. However, in some cases fractional solutions are unrealistic. We have to consider the integer number for those decision variables. For example, number of worker, number of batches per day or month and number of heat exchanger in any HEN.

The optimization problem can be written for these systems

$$\text{Minimize } f = \sum_{j=1}^{n} c_j x_j \tag{6.36a}$$

subject to

$$\sum_{j=1}^{n} a_{ij} x_j = b_i, \ \left(i = 1, 2, \ldots, m;\ j = 1, 2, \ldots, \mathrm{n}\right) \tag{6.36b}$$

$$x \in Z_+^n \tag{6.36c}$$

where Z_+^n denotes the set of dimensional vectors n having integer non-negative components.

The problem of Eq. (6.36a)–(6.36c) is called linear integer programming problem. When, a few variables but not all are restricted to be an integer, it is said to be a mixed integer program and is termed a pure integer program whilst all decision variables must be integers. Shaded area in Fig. 6.8 is the feasible region when variables (x_1, x_2) are real numbers, whereas red dots represents the feasible points for integer linear programming. Now, we will discuss two useful Mixed Integer Linear Programming (MILP) models.

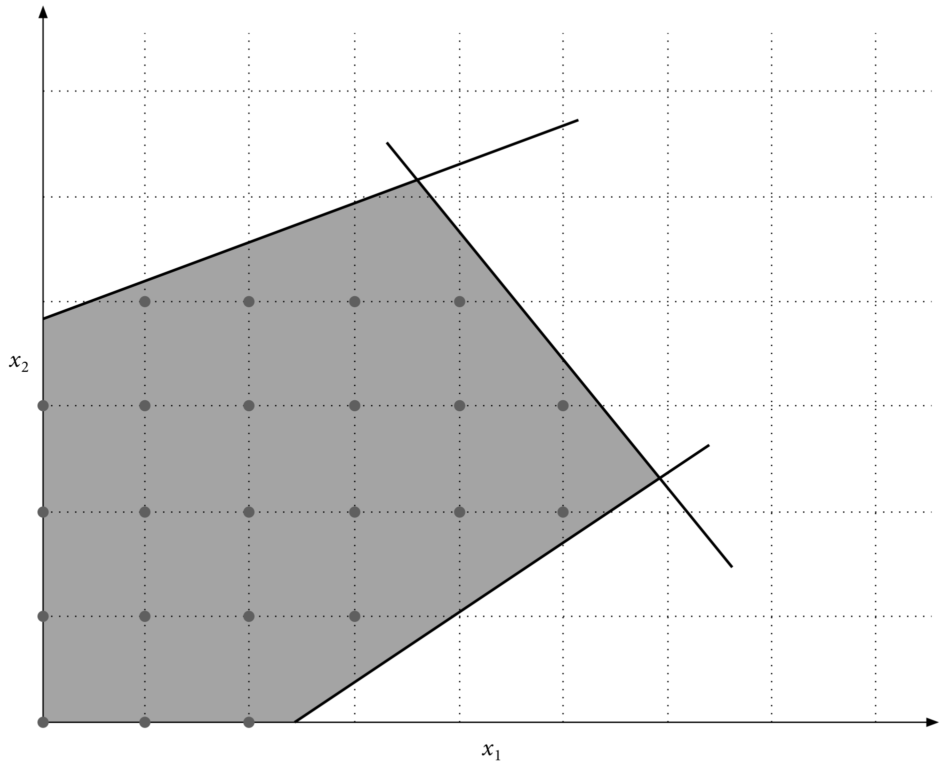

Fig. 6.8 Integer linear programming

Example 6.4 Warehouse Location During modeling any distribution system, a trade off between cost of transportation and operating cost of distribution centers is required for taking any decision. Say for example, a marketing manager required to take decision on which of n warehouses to be used that will meet the demands of m customers for a substance. The decisions to be made are which warehouses to be open and how much to ship from any warehouse to any customer. Let,

$$\text{Minimize } f = \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij} + \sum_{i=1}^{m} f_i y_i \tag{6.37a}$$

subject to

$$\sum_{i=1}^{m} x_{ij} = d_j \ (j = 1, 2, \ldots, n) \tag{6.37b}$$

$$\sum_{j=1}^{n} x_{ij} - y_i\left(\sum_{j=1}^{n} d_j\right) \le 0 \ \ (i = 1, 2, \ldots, m) \tag{6.37c}$$

$$x_{ij} \geq 0 \;\; (.., j = 1, 2, \ldots, \mathrm{n}) \tag{6.37d}$$

$$y_i = 0 \;\text{ or }\; 1 \;\; (i = 1, 2, \ldots, m) \tag{6.37e}$$

where

x_{ij} = Amount to be sent from warehouse i to customer j.

$$y_i = \begin{cases} 1 & \text{if warehouse } i \text{ is opened} \\ 0 & \text{if warehouse } i \text{ is not opened} \end{cases}$$

f_i = Fixed operating cost for warehouse i, if opened (for example, a cost to lease the warehouse),

c_{ij} = Per-unit operating cost at warehouse i plus the transportation cost for shipping from warehouse i to customer j.

Here, we have considered

i. the warehouses must fulfill the demand d_j of each customer; and
ii. shipment of goods from a warehouse is possible only if it is opened.

Example 6.5 Blending problem We have to prepare a blend from a given list of ingredients. The list gives us various information such as weight, value, cost, and analysis of each ingredient [Edger *et al.* (2001)].

Our objective is to prepare a blend of some specified total weight with satisfactory analysis by selecting a set of ingredients from the list with a minimum cost for a blend. Let, x_j is the amount of ingredient j available (continuous amounts) and y_k indicate ingredients to be utilized in discrete quantities v_k (if it is used y_k = 1and y_k = 0 if not used). The quantities c_j and d_k be the respective costs of the ingredients and a_{ij} be the fraction of the component i in ingredients j. The problem can be stated is

$$\text{Minimize } f = \sum_j c_j x_j + \sum_k d_k v_k y_k \tag{6.38a}$$

subject to:

$$W^l \leq \sum_j x_j + \sum_k v_k y_k \leq W^u \tag{6.38b}$$

$$A_i^l \leq \sum_j a_{ij} x_j + \sum_k a_{ik} v_k y_k \leq A_i^u \tag{6.38c}$$

$$0 \leq x_j \leq u_j \;\text{ for all } j \tag{6.38d}$$

$$y_k = (0,1) \;\text{ for all } k \tag{6.38e}$$

where, u_j = upper limit of the jth ingredient,

W^u and W^l = the upper and lower bounds on the weights respectively

A_i^u and A_i^l = the upper and lower bounds on the analysis for component i respectively

The following section elucidates the Branch and Bound method for solving MILP.

Branch-and-Bound method

Branch-and-Bound method has been developed by Land and Doig [Land and Doig (1960)]. This method is very efficient for solving mixed-integer linear and nonlinear programming. Feroymson and Ray used this method for plant location problem [Feroymson and Ray, (1966)]. This method consists of two basic operations. Branching, which divide the whole set of solution into subsets and bounding that consists of establishing bounds on the value of the objective function over the subsets. The branch-and-bound method involves recursive application of the branching and bounding operations, with a provision for eliminating subsets that do not contain any optimal solution [L. G. Mitten (1970)].

The integer problem is not directly solved in branch-and-bound method. Whereas, it relaxes the integer restrictions on the variables and solves as a continuous problem [Rao (2001)]. Sometimes the solution of that continuous problem may be an integer solution; on that case, we can consider it as the optimum solution of the integer problem. Otherwise, we have to assume at least one of the integer variables, say x_i, must have a nonintegral value. Whenever x_i is not an integer, always we can find an integer $[x_i]$ such that,

$$[x_i] < x_i < [x_i] + 1 \tag{6.39}$$

Then, we have to formulate two subproblems, one with the additional upper bound constraint

$$x_i \le [x_i] \tag{6.40}$$

and another with lower bound as the additional constraint

$$x_i \ge [x_i] + 1 \tag{6.41}$$

This method of establishing the subproblems is called branching.

The branching operation removes some part of the continuous space, which is not feasible for the integer problem. At the same time, we have to ensure that any of the integer feasible solutions is not discarded. Then we have to solve these two subproblems as continuous problems. It is observed that the solution to the continuous problem forms a node and two branches may originate from this node.

This practice of branching and solving a series of continuous problems is continued until an integer feasible solution is obtained for one of the two continuous problems. Whenever such a feasible integer solution is obtained, the objective function value at that point turns into an upper bound on the minimum value of the objective function. At this instant, further consideration is not required for all these continuous solutions (nodes) that have objective function values larger than the upper bound. The eliminated nodes are said to have been fathomed since it is impossible to get a better integer solution from these nodes (solution spaces) than whatever we have at the present. Whenever a better bound is obtained, then the value of the upper bound on the objective function is updated [Rao (2001)]. This process can be illustrated by the example 6.6.

Example 6.6

Solve the problem using branch and bound method.

Maximize: $Z = 5x_1 + 7x_2$

subject to constraints

$$x_1 + x_2 \leq 6$$

$$5x_1 + 9x_2 \leq 43$$

$x_1, x_2 \geq 0$ and Integer

Solution

At the first step, we solve the problem as a continuous problem. The corresponding result is

$$x_1 = 2.75, \ x_2 = 3.25 \text{ and } Z = 36.50$$

As x_1, x_2 are not integer, we have to form two subproblems (Eqs (6.40), (6.41))

one with additional upper bound $x_1 \leq 2$ (subproblem 1) and another is with lower bound $x_1 \geq 3$ (subproblem 2)

Table 6.1 Subproblems for branch and bound method

Subproblem 1	Subproblem 2
Maximize: $Z = 5x_1 + 7x_2$ subject to constraints	Maximize: $Z = 5x_1 + 7x_2$ subject to constraints
$x_1 + x_2 \leq 6$	$x_1 + x_2 \leq 6$
$5x_1 + 9x_2 \leq 43$	$5x_1 + 9x_2 \leq 43$
$x_1 \leq 2$	$x_1 \geq 3$
$x_1, x_2 \geq 0$	$x_1, x_2 \geq 0$

By solving these subproblems, we get

Table 6.2 Results of the subproblems in Table 6.1

Solution of subproblem 1	Solution of subproblem 2
$x_1 = 2$	$x_1 = 3$
$x_2 = 3.667$	$x_2 = 3$
$Z_1^* = 35.667$	$Z_2^* = 36$

Here, subproblem 2 is fathomed as x_1, x_2 are integer.

As $Z_1^* \le Z_2^*$, we can stop branching. Then, the final solution is

$x_1^* = 3$

$x_2^* = 3$

$Z^* = 36$

Note The code for solving this problem using LINGO is given in chapter 12.

6.3 Nonlinear Programming with Constraints

In chapter 3 and 5, we have discussed various methods of nonlinear optimization without any constraint. Optimization with constraints is very common in chemical engineering. Most of the chemical process plant faces the limitations for raw materials, manpower, utilities, and space. We can include these constraints during optimization of the process variables. Another constraints might appear due to environmental considerations. The following sections elucidate the constraints nonlinear optimization

6.3.1 Problems with equality constraints

The constrained optimization problems with equality constraints could be represented as the following:

$$\text{Minimize } f(X) \quad X \in \mathbb{R}^n \tag{6.42a}$$

$$\text{subject to } c_i(X) = 0 \tag{6.42b}$$

$$X \ge 0$$

where $f(X)$ is the objective function, $c_i(X)$ are constraint functions.

6.3.1.1 Direct substitution

For a problem with m equality constraints consist of n variables, theoretically we are able to solve these m equality constraints simultaneously and represent any set of m variables in terms of the remaining $n - m$ variables. The substitution of these expressions into the original objective function gives us a new objective function that involves only $n - m$ variables. This new objective function becomes an unconstrained problem that is not subjected to any constraint. Therefore, optimization of this objective function can be performed by using the unconstrained optimization techniques discussed in Chapter 5.

$$\text{Minimize: } f(X) = 2x_1^2 + 2x_2^2 + x_3^2 - 2x_1x_2 - 4x_1 - 6x_2 \tag{6.43a}$$

$$\text{subject to: } x_1 + x_2 + x_3 = 2 \tag{6.43b}$$

$$x_1^2 + 5x_2 = 5 \tag{6.43c}$$

We will consider this problem to elucidate the "Direct Substitution Method". In the first step, Eq. (6.43c) is rearranged as

$$x_2 = \frac{5 - x_1^2}{5} \tag{6.44}$$

Now, replace x_2 from Eq. (6.43a) and (6.43b) using Eq. (6.44).

after rearrangement the final form is

$$\text{Minimize: } f(X) = \frac{2}{25}x_1^4 + \frac{12}{5}x_1^2 + x_1^3 + \frac{2}{5}x_1^2 x_3^2 - 6x_1 - 4 \tag{6.45a}$$

$$\text{subject to: } -x_1^2 + 5x_1 + 5x_3 - 5 = 0 \tag{6.45b}$$

this problem has 2 variables and 1 equality constraint

Again, we have to substitute x_3 from Eq. (6.45b) to Eq. (6.45a). Then, the final equation will be

$$\text{Minimize: } f(X) = \frac{2}{25}x_1^4 + \frac{12}{5}x_1^2 + x_1^3 + \frac{2}{125}x_1^2\left(5 - 5x_1 + x_1^2\right)^2 - 6x_1 - 4 \tag{6.46}$$

Equation (6.46) is an unconstrained optimization problem. We will discuss the problem formulated in section 2.3.1.

Example 6.7

The price per unit side area is \$5 and price per unit area of top and bottom is \$8. Find the optimum diameter and height of the for a 1000 liter tank.

$$\min_{L,D} f = c_s \pi DL + c_t (\pi/2) D^2 \tag{2.1}$$

$$\text{subject to } V = (\pi/4) D^2 L \tag{2.2}$$

Solution

Incorporating the values $c_s = 5$ and $c_t = 8$ in Eq. (2.1), we get

$$\min_{L,D} f = 15.7DL + 12.56D^2 \tag{6.47a}$$

$$\text{subject to } D^2 L = 1273.9 \tag{6.47b}$$

and $V = 1000$ (6.47c)

We can use "Direct substitution method" to solve this problem. For this purpose Eq. (6.47b) can be written as

$$L = \frac{1273.9}{D^2} \tag{6.48}$$

Now substitute L to the Eq. (6.47a), to get

$$\min_{D} f = \frac{20000.23}{D} + 12.56D^2 \tag{6.49}$$

The Eq. (6.49) is an unconstrained problem with 2 – 1 = 1 variable. Now this problem can be solved by using any method described in chapter 3.

6.3.1.2 Lagrange multiplier method

Lagrange multiplier method can be used for problems with equality constraints. The Lagrange multiplier method is discussed by the Example 6.8 of two variables with one constraint. The form of a general problem with n variables and m constraints is given later.

Consider a problem as follows:

Minimize $f(x_1, x_2)$ (6.50a)

subject to $g(x_1, x_2) = 0$ (6.50b)

The necessary condition for the existence of an extreme point at $X = X^*$

$$\left(\frac{\partial f}{\partial x_1} - \frac{\partial f/\partial x_2}{\partial g/\partial x_2}\frac{\partial g}{\partial x_1}\right)\Bigg|_{(x_1^*, x_2^*)} = 0 \tag{6.51}$$

The Lagrange multiplier (λ) can be defined as

$$\lambda = -\left(\frac{\partial f/\partial x_2}{\partial g/\partial x_2}\right)\Bigg|_{(x_1^*, x_2^*)} \tag{6.52}$$

The Eq. (6.51) can be expressed as

$$\left(\frac{\partial f}{\partial x_1} + \lambda\frac{\partial g}{\partial x_1}\right)\Bigg|_{(x_1^*, x_2^*)} = 0 \tag{6.53}$$

and Eq. (6.52) can be expressed as

$$\left(\frac{\partial f}{\partial x_2}+\lambda\frac{\partial g}{\partial x_2}\right)\Bigg|_{\left(x_1^*,x_2^*\right)}=0 \tag{6.54}$$

In addition, the constraint equation should satisfy the extreme point, that is,

$$g\left(x_1,x_2\right)\Big|_{\left(x_1^*,x_2^*\right)}=0 \tag{6.55}$$

These are the necessary conditions for the point $\left(x_1^*,x_2^*\right)$ to be an extreme point.

Notice that the partial derivative $\left(\partial g/\partial x_2\right)\Big|_{\left(x_1^*,x_2^*\right)}$ has to be nonzero to be able to define λ by Eq. (6.52). This is because the variation dx_2 was expressed in terms of dx_1. On the other hand, if dx_1 is expressed in terms of dx_2, we would have obtained the requirement that $\left(\partial g/\partial x_1\right)\Big|_{\left(x_1^*,x_2^*\right)}$ be nonzero to define λ. Therefore, the derivation of the necessary conditions using the method of Lagrange multipliers requires that at least one of the partial derivatives of $g(x_1, x_2)$ be nonzero at an extreme point.

Construct a function L, known as Lagrange function

$$L\left(x_1,x_2,\lambda\right)=f\left(x_1,x_2\right)+\lambda g\left(x_1,x_2\right) \tag{6.56}$$

by treating L as a function of the three variables x_1, x_2 and λ, the necessary conditions for its extremum are given by

$$\frac{\partial L}{\partial x_1}\left(x_1,x_2,\lambda\right)=\frac{\partial f}{\partial x_1}\left(x_1,x_2\right)+\lambda\frac{\partial g}{\partial x_1}\left(x_1,x_2\right)=0 \tag{6.57}$$

$$\frac{\partial L}{\partial x_2}\left(x_1,x_2,\lambda\right)=\frac{\partial f}{\partial x_2}\left(x_1,x_2\right)+\lambda\frac{\partial g}{\partial x_2}\left(x_1,x_2\right)=0 \tag{6.58}$$

$$\frac{\partial L}{\partial \lambda}\left(x_1,x_2,\lambda\right)=g\left(x_1,x_2\right)=0 \tag{6.59}$$

Sufficiency conditions: A sufficient condition for $f(X)$ to have a constrained relative minimum at X^* is given by the following theorem.

Theorem 6.1

A sufficient condition for $f(X)$ to have relative minimum at X^* is that the quadratic, Q, defined by

$$Q=\sum_{i=1}^{n}\sum_{j=1}^{n}\frac{\partial^2 L}{\partial x_i\partial x_j}dx_i dx_j \tag{6.60}$$

evaluate at $X = X^*$ must be positive definite for all values of dX for which the constraints are satisfied.

It has been shown by Hancock [Hancock, (1960)] that a necessary condition for the quadratic form Q in Eq. (6.60), to be positive (negative) definite for all admissible variations dX is that each root of the polynomial z_i, defined by the following determinantal equation, be positive (negative):

$$\begin{vmatrix} L_{11}-z & L_{12} & L_{13} & \cdots & L_{1n} & g_{11} & g_{21} & \cdots & g_{m1} \\ L_{21} & L_{22}-z & L_{21} & \cdots & L_{21} & g_{12} & g_{22} & \cdots & g_{m2} \\ \vdots & & & & & & & & \\ L_{n1} & L_{n2} & L_{n3} & \cdots & L_{nn}-z & g_{1n} & g_{2n} & \cdots & g_{mn} \\ g_{11} & g_{12} & g_{13} & \cdots & g_{1n} & 0 & 0 & \cdots & 0 \\ g_{21} & g_{22} & g_{23} & \cdots & g_{2n} & 0 & 0 & \cdots & 0 \\ \vdots & & & & & & & & \\ g_{m1} & g_{m2} & g_{m3} & \cdots & g_{mn} & 0 & 0 & \cdots & 0 \end{vmatrix} = 0 \tag{6.61}$$

where

$$L_{ij} = \frac{\partial^2 L}{\partial x_i \partial x_j}\left(X^*, \lambda^*\right) \tag{6.62}$$

$$g_{ij} = \frac{\partial g_i}{\partial x_j}\left(X^*\right) \tag{6.63}$$

Equation (6.61) is an $(n - m)$th order polynomial in z. The point X^* is not an extreme point when some of the roots of this polynomial are positive whereas the others are negative.

Example 6.8

Minimize the function using Lagrange multiplier method

Minimize $f\left(x_1, x_2\right) = 25 + x_1 - 3x_2 + x_1^2 + 2x_2^2 - 5x_1x_2$

subject to $x_1^2 + x_2^2 = 7$

Solution

The Lagrange function $L\left(x_1, x_2, \lambda\right) = \left(25 + x_1 - 3x_2 + x_1^2 + 2x_2^2 - 5x_1x_2\right) + \lambda\left(x_1^2 + x_2^2 - 7\right)$

The necessary conditions

$$\frac{\partial L}{\partial x_1} = 1 + 2x_1 - 5x_2 + \lambda\left(2x_1\right) = 0$$

$$\frac{\partial L}{\partial x_2} = -3 + 4x_2 - 5x_1 + \lambda\left(2x_2\right) = 0$$

$$\frac{\partial L}{\partial \lambda} = x_1^2 + x_2^2 - 7 = 0$$

Solving these equations, we get

$$x_1^* = 1.8585,\ x_2^* = 1.8831,\ \text{and}\ \lambda^* = 1.2641$$

which gives

$$f^* = 14.25676$$

To check whether this solution really corresponds to the minimum of *f*, we can apply the sufficiency condition.

$$L_{11} = \left.\frac{\partial^2 L}{\partial x_1^2}\right|_{(X^*,\lambda^*)} = 2 + 2\lambda^* = 4.5282$$

$$L_{12} = \left.\frac{\partial^2 L}{\partial x_1 \partial x_2}\right|_{(X^*,\lambda^*)} = -5$$

$$L_{22} = \left.\frac{\partial^2 L}{\partial x_2^2}\right|_{(X^*,\lambda^*)} = 4 + 2\lambda^* = 6.5282$$

$$g_{11} = \left.\frac{\partial g_1}{\partial x_1}\right|_{(X^*,\lambda^*)} = 2x_1^* = 3.717$$

$$g_{12} = \left.\frac{\partial g_1}{\partial x_2}\right|_{(X^*,\lambda^*)} = 2x_2^* = 3.7662$$

equation (6.61) can be written as

$$\begin{vmatrix} 4.5282 - z & -5 & 3.717 \\ -5 & 6.5282 - z & 3.7662 \\ 3.717 & 3.7662 & 0 \end{vmatrix} = 0$$

which gives $z = 10.51462$. This result confirms that the f^* is the minimum value.

6.3.2 Problems with inequality constraints

The general form of an optimization with inequality constraints is given below

$$\text{Minimize } f(X) \qquad X \in \mathbb{R}^n \tag{6.64a}$$

$$\text{subject to } c_j(X) \geq 0, \quad j = 1, 2, \ldots m \tag{6.64b}$$

$$X \geq 0 \tag{6.64c}$$

where $f(X)$ is the objective function and $c_j(X) \geq 0$ are inequality constraints. Lagrange multiplier method can be used to solve these problems.

An inequality constraint $c_j(X) \geq 0$ is said to be active at X^* if $c_j(X^*) \geq 0$. It is inactive at X^* if $c_j(X^*) > 0$ [Chong and Zak, (2001)].

6.3.2.1 Kuhn–Tucker condition

If the problem has inequality constraints, the Kuhn–Tucker condition can be used to identify the optimum point. However, these methods produce a set of nonlinear simultaneous equations that may be difficult to solve.

The nonlinear programming (NLP) problem with one objective function $f(X)$ and m constraint functions c_j which are continuously differentiable, can be represented as follows:

$$\text{Minimize } f(X), \; X \in \Re^n \tag{6.65a}$$

$$\text{subject to } c_j(X) \leq 0, \quad j = 1, 2, \ldots m \tag{6.65b}$$

In the preceding notation, n is the dimension of the function $f(X)$, and m is the number of inequality constraints.

$$L(X, \lambda) = f(X) - \sum_{j=1}^{m} \lambda_j c_j(X) \tag{6.66}$$

is the Lagrange function, and the coefficients λ_j are the Lagrange multipliers.

If the functions $f(X)$ and c_j are twice differentiable, the point X^* is an isolated local minimizer of the NLP problem, if there exists a vector $\lambda^* = \left(\lambda_1^*, \lambda_2^*, \ldots, \lambda_m^*\right)$ that meets the following conditions:

$$\frac{\partial f}{\partial x_i} + \sum_{j \in J_1} \lambda_j \frac{\partial c_j}{\partial x_i} = 0, \quad i = 1, 2, \ldots n \tag{6.67a}$$

$$\lambda_j > 0, \quad j \in J_1 \tag{6.67b}$$

For convex programming, the Kuhn–Tucker conditions are necessary and sufficient for a global minimum. The Kuhn–Tucker conditions can be stated as follows:

$$\frac{\partial f}{\partial x_i} + \sum_{j=1}^{m} \lambda_j \frac{\partial g_j}{\partial x_i} = 0, \quad i = 1, 2, \ldots n \tag{6.68a}$$

$$\lambda_j g_j = 0,\ j = 1,2,\ldots m \tag{6.68b}$$

$$g_j \le 0,\ j = 1,2,\ldots m \tag{6.68c}$$

$$\lambda_j \ge 0,\ j = 1,2,\ldots m \tag{6.68d}$$

6.3.2.2 Logarithmic barrier method

During optimization process, the algorithm may cross the boundary of feasible region. We can follow a technique to prevent the optimization algorithm from crossing the boundary by assigning a penalty to approaching it. The most accepted way of doing this is to augment the objective function by a logarithmic barrier term:

$$B(x,\mu) = f(x) - \mu \sum_{i=1}^{p} \log\left(h_i(x)\right) \tag{6.69}$$

Here, log denotes the natural logarithm. Since,

$$-\log(t) \to \infty \text{ as } t \to 0, \tag{6.70}$$

The term $B(x,\mu)$ "blows up" at the boundary and consequently presents an optimization algorithm with a "barrier" to crossing the boundary. Certainly, the solution to an inequality-constrained is expected to lie on the boundary of the feasible set, so the barrier should be eliminated gradually by decreasing the value of μ toward zero. The following strategy has been suggested:

Choose $\mu_0 > 0$ and a strictly feasible point $x^{(0)}$.

For $k = 1, 2, 3, \ldots$

Choose $\mu_k \in (0, \mu_{k-1})$ (perhaps $\mu_k = \beta\mu_{k-1}$ for some constant $\beta \in (0,1)$.

Using $x^{(k-1)}$ as the starting point, solve

$$\min B(x, \mu_k) \tag{6.71}$$

to get $x^{(k)}$

Under certain conditions, $B(x,\mu)$ has a unique minimizer x_μ^* in a neighbourhood of x^* and that $x_\mu^* \to x^*$ as $\mu \to 0$.

Then for all μ sufficiently small

$$\nabla B\left(x_\mu^*, \mu\right) = 0 \tag{6.72}$$

6.3.3 Convex optimization problems

Problems involving the optimization of convex functions (objective as well as constraints functions) are called convex optimization problems. This class of problems has some advantages over other form of optimization problems; local optimizer is the global optimizer for them.

The standard form of a convex optimization problem

$$\text{Minimize } f_0(X) \tag{6.73a}$$

$$\text{subject to } f_i\left(X\right) \le 0,\ i = 1,2,\ldots m \tag{6.73b}$$

$$h_i\left(X\right) = 0,\ i = 1,2,\ldots p \tag{6.73c}$$

where

$X \in R^n$ is the optimization variable (convex)

$f_0 : R^n \to R$ is the objective or cost function (convex function)

$f_i : R^n \to R,\ i = 1,2,\ldots m$, are the inequality constraint functions (convex function)

$h_i : R^n \to R$ are equality constraint functions (affine functions)
optimal value of the problem (6.73a–6.73c) are as follows:

$$p^* = \inf\left\{f_0\left(X\right) \middle| f_i\left(X\right) \le 0,\ i = 1,\ldots,m,\ h_i\left(X\right) = 0,\ i = 1,\ldots,p\right\} \tag{6.74a}$$

$$p^* = \infty \text{ if problem is infeasible (no } X \text{ satisfies the constraints)} \tag{6.74b}$$

$$p^* = -\infty \text{ if problem is unbounded below} \tag{6.74c}$$

i. X is a feasible if $X \in \operatorname{dom} f_0$ and it satisfies the constraints and a feasible X is optimal if $f_0\left(X\right) = p^*$; X_{opt} is the set of optimal points.
ii. X is locally optimal if there is an $R > 0$ such that X is optimal for

$$\text{Minimize (over } z) \ f_0\left(z\right) \tag{6.75a}$$

$$\text{subject to } f_i\left(z\right) \le 0,\ i = 1,2,\ldots m;\ h_i\left(z\right) = 0,\ i = 1,2,\ldots p \tag{6.75b}$$

$$\left\|z - X\right\|_2 \le R \tag{6.75c}$$

The problem is quasiconvex when $f(X)$ is quasiconvex (and $g_i(X)$ are convex) often written as

$$\text{Minimize } f(X) \tag{6.76a}$$

$$\text{subject to } g_i\left(X\right) \le 0,\ i = 1,2,\ldots m \tag{6.76b}$$

$$AX = b,\ i = 1,2,\ldots p \tag{6.76c}$$

important property: feasble set of a convex optimization problem is convex

Optimality condition

Considering the optimization problem above, which we write as

$$\min_{x \in X} f_0(x) \tag{6.77}$$

where X is the feasible set.

When f_0 is differentiable, then we know that for every $x, y \in \operatorname{dom} f_0$,

$$f_0(y) \geq f_0(x) + \nabla f_0(x)^T (y - x) \tag{6.78}$$

Then, x is optimal if and only if

$$x \in X \text{ and } \forall y \in X : \nabla f_0(x)^T (y - x) \geq 0 \tag{6.79}$$

If $\nabla f_0(x) \neq 0$, then it defines a supporting hyperplane to the feasible set at x.

When, the problem is unconstrained, we obtain the optimality condition:

$$\nabla f_0(x) = 0 \tag{6.80}$$

Note that these conditions are not always feasible, since the problem may not have any minimizer. This can happen for example when the optimal value is only attained in the limit; or, in constrained problems, when the feasible set is empty.

Theorem 6.2

Any locally optimal point of a convex problem is (globally) optimal

Proof

Let x^* be a local minimizer of f_0 on the set X, and let $y \in X$. By definition, $x^* \in \operatorname{dom} f_0$. We need to prove that $f_0(y) \geq f_0(x^*) = p^*$. There is nothing to prove if $f_0(y) = +\infty$, so let us assume that $y \in \operatorname{dom} f_0$. By convexity of f_0 and X, we have $x_\theta := \theta y + (1 - \theta) x^* \in X$, and:

$$f_0(x_\theta) - f_0(x^*) \leq \theta\big(f_0(y) - f_0(x^*)\big) \tag{6.81}$$

as x^* is a local minima, the left-hand side in this inequality is nonnegative for all small enough values of $\theta > 0$. We conclude that the right-hand side is nonnegative, i.e., $f_0(y) \geq f_0(x^*)$, as claimed.

Also, the optimal set is convex, since it can be written

$$X^{\text{opt}} = \left\{x \in R^n : f_0(x) \leq p^*,\ x \in X\right\} \tag{6.82}$$

This proofs the statement of the theorem.

Summary

- Formulation and solution methods for multivariable and constrained optimization have been considered in this chapter. Both the methods of linear and nonlinear programming algorithms are discussed with proper examples. Integer linear programming that is extensively applicable for distribution of chemical products from different warehouses to different distributors and the blending process are considered. This chapter also enlightens the theories and examples of nonlinear constrained optimization processes.

  Further studies Real-time optimization of the pulp mill benchmark problem; Mehmet Mercangöz, Francis J. Doyle III; Computers and Chemical Engineering 32 (2008) 789–804

Review Questions

6.1 Is it possible to convert a constrained optimization problem to an unconstrained optimization problem? Explain your answer with proper example.

6.2 Maximize $f = 5x_1 + 3x_2$

subject to the constraints

$$2x_1 + x_2 \le 1000$$

$$x_1 \le 400$$

$$x_2 \le 700$$

$$x_1, x_2 \ge 0$$

6.3 Convert the following problem to its dual form

Maximize $f = 400x_1 + 200x_2$

subject to the constraints

$$18x_1 + 3x_2 \le 800$$

$$9x_1 + 4x_2 \le 600$$

$$x_1, x_2 \ge 0$$

6.4 A refinery distills two crude petroleum, A and B, into three main products: jet fuel, gasoline and lubricants. The two crudes differ in chemical composition and thus, yield different product mixes (the remaining 10 per cent of each barrel is lost to refining): Each barrel of crude A yields 0.4 barrel of jet fuel, 0.3 barrel of gasoline, and 0.2 barrel of lubricants; Each barrel of crude B yields 0.2 barrel of jet fuel, 0.4 barrel of gasoline, and 0.3 barrel of lubricants. The crudes also differ in cost and availability: Up to 5,000 barrels per day of crude A are available at the cost \$25 per barrel; Up to 3,000 barrels per day of Saudi crude are also available at the lower cost \$18 per barrel.

Contracts with independent distributors require that the refinery produce 2,000 barrels per day of gasoline, 1,500 barrels per day of jet fuel, and 500 barrels per day of lubricants.

6.5 What are the advantages of Khachiyan's ellipsoid method over simplex method?

6.6 Minimize $f = -3x_1 + 2x_2$

Subject to $x_1 + x_2 \le 9$

$x_2 \le 6$

$x_1, x_2 \ge 0$

make the "off-center" point to an "equidistant" from the coordinate axes in a transformed feasible region by "variable scaling".

6.7 Solve the following problem by branch and bound method.

Maximize $f = x_1 + x_2$

subject to the constraints

$2x_1 + 5x_2 \le 16$

$6x_1 + 5x_2 \le 30$

$x_1, x_2 \ge 0$ and Integer

6.8 Solve the problem using direct substitution method

Maximize $f = 3x_1^2 + 2x_2^2$

subject to the constraints

$x_1 + 3x_2 = 5$

6.9 Solve the problem using Lagrange multiplier method

Minimize $f = 3x_1^2 + 4x_2^2 + x_1x_3 + x_2x_3$

subject to

$x_1 + 3x_2 = 3$

$2x_1 + x_3 = 7$

6.10 Find the minimum value of the function

$f = x_1^2 + x_2^2 - x_1x_2$

subject to the inequality constraint

$x_1 + 3x_2 \ge 3$

6.11 Why convex optimization problem is easier to solve compare to nonconvex problem?

6.12 We have 25 ft steel frame for manufacturing a window. The shape of window is shown in Figure. Estimate the optimum dimensions such that the area of the window will be maximum.

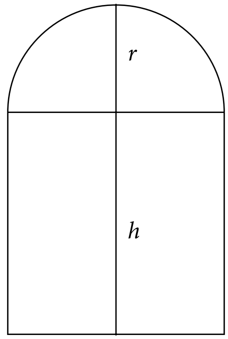

Fig. 6.9 Shape of the window

References

Bland, R. G., Goldfarb, D., Todd, M. J. 1981. *Feature Article-The Ellipsoid Method:* A Survey, Operations Research, 29(6): 1039–91.

Chong, E. K. P., Zak, S. H. 2001. *An Introduction to Optimization Second Edition,* John Wiley and Sons, Inc.

Dantzig, G. B. 1963. *Linear Programming and Extensions*, Princeton University Press.

Dantzig, G. B. 2002. *Linear Programming, Operations Research,* vol. 50, No. 1, January–February 2002, pp. 42–47.

Edgar, T. F., Himmelblau, D. M., Lasdon, L. S. 2001. *Optimization of Chemical Processes* McGraw–Hill.

Feroymson, M. A. and Ray, T. L. 1966. *A Branch-and-Bound Algorithm for Plant Location*, Opns Res 14, 361–68.

Fletcher, R. *Practical Methods of Optimization*, John Wiley and Sons.

Hancock, H., 1960. *Theory of Maxima and Minima*, New York: Dover.

Karmarkar, N. K. 1984. *'A New Polynomial Time Algorithm for Linear Programming',* Combinatorica, 4, 373–95.

Khachiyan, L. G., 1979. *A Polynomial Algorithm in Linear Programming.* Doklady Akademiia Nauk SSSR, 244: 1093–96, English Translation: Soviet Mathematics Doklady, 20(1): 191–94,.

Land, A. H. and Doig, A. G. *An Automatic Method of Solving Discrete Programming Problems,* Econometrica, vol. 28, No. 3. (Jul., 1960), pp. 497–520.

Lemire, N. 1989. *An Introduction to Karmarkar's Method,* Windsor Mathematics Report WMR 89–10; July, 1989.

Mitten, L. G. 1970. *Branch-and-Bound Methods: General Formulation and Properties,* Operations Research, vol. 18, No. 1, pp. 24–34.

Rao, S. S. 2009. *Engineering Optimization: Theory and Practice (4e),* Wiley and Sons, Inc.

Ravindran, A., Ragsdell, K. M., Reklaitis, G. V. 2006. *Engineering Optimization: Methods and Applications*, John Wiley and Sons, Inc.

Rebennack, S. 2008. *Ellipsoid Method, Encyclopedia of Optimization, Second Edition*, C. A. Floudas and P. M. Pardalos (Eds.), Springer, pp. 890–99.